BLACK CODHER

CODING PROGRAMME

<CODING BLACK FEMALES/>

Black Codher Bootcamp

NÍ YỌ

BLACK CODHER

CODING PROGRAMME

UNIT 3
Javascript 106

YOUR LAST JAVASCRIPT SESSION

You've persevered for 3 weeks and now you're only 3 hours away from the end of the JavaScript unit. Let the countdown begin

WHAT YOU'LL BE LEARNING

- Modules

MODULES

WHAT IS A MODULE?

If we think about books, they are divided into chapters and section. In Javascript a program is divided into module. Each module is a single file. it's as simple as that.

Good modules, however, are highly self-contained with distinct functionality, allowing them to be shuffled, removed, or added as necessary, without disrupting the system as a whole.

WHY USE MODULES?

There are many benefits to using modules, and they are great for organising your code. Having one file for your entire website/application will become very messy and confusing very quickly.

Imagine you had 30 functions in one file, and one of them was missing a semicolon that caused an error 😱. Looking through that file could be a headache.

WHY USE MODULES?

Let's get into the modules mindset and think of something we do everyday and how we can break it into separate modules: *getting ready for bed*.

We all have different routines, but generally we:

- Brush teeth
- Do some sort of skincare
- Put on PJs
- Sort out hair
- Turn off lights
- Get into bed
- Close eyes
- Sleep

WHY USE MODULES?

We could create one module that does all of these things OR we could break them up into different modules. By breaking them up separately we can use them all independently or group them together or reuse them for other routines.

If you want to have an afternoon nap you don't need to brush your teeth before hand.

BENEFITS OF MODULES

Maintainability

A module should be self-contained, and if well-designed creates less dependencies on the rest of the codebase. Updating a single module is much simpler when decoupled from other pieces of code. E.g. a brand new skincare routine should not interfere with turning off the lights

Reusability

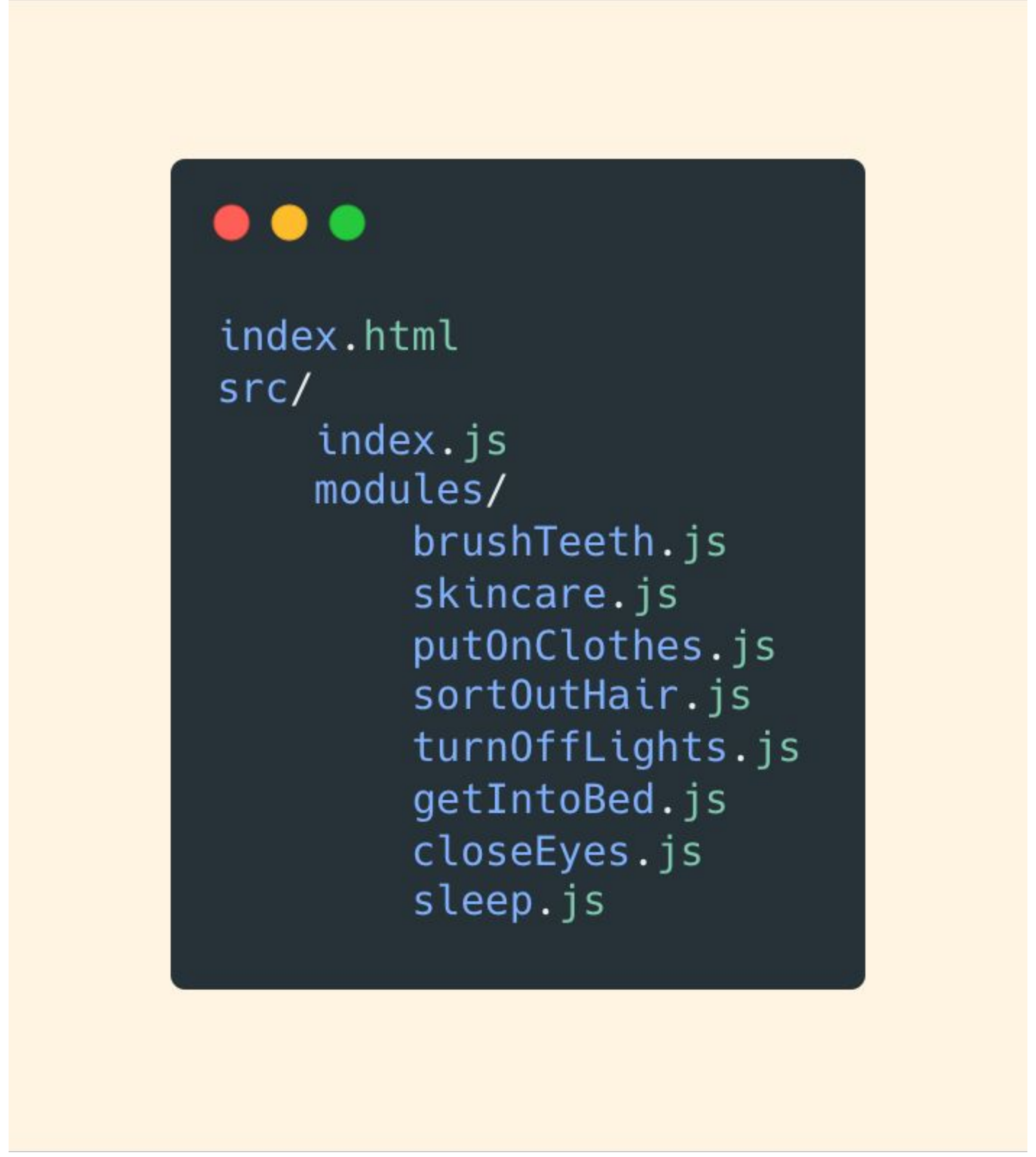
During the HTML/CSS unit you heard about DRY (don't repeat yourself), and not many of us like having to repeat ourselves, right? Say we want to create a morning routine, we could definitely reuse *brush teeth, do some sort of skincare* and *sort out hair.*

HOW TO USE MODULES

In JavaScript there are many ways to use modules. The most common way is using `import` and `export`. Let's build up our *getting ready for bed* and a *get ready for the day* routine**.**

HOW TO USE MODULES

Our file structure will be:

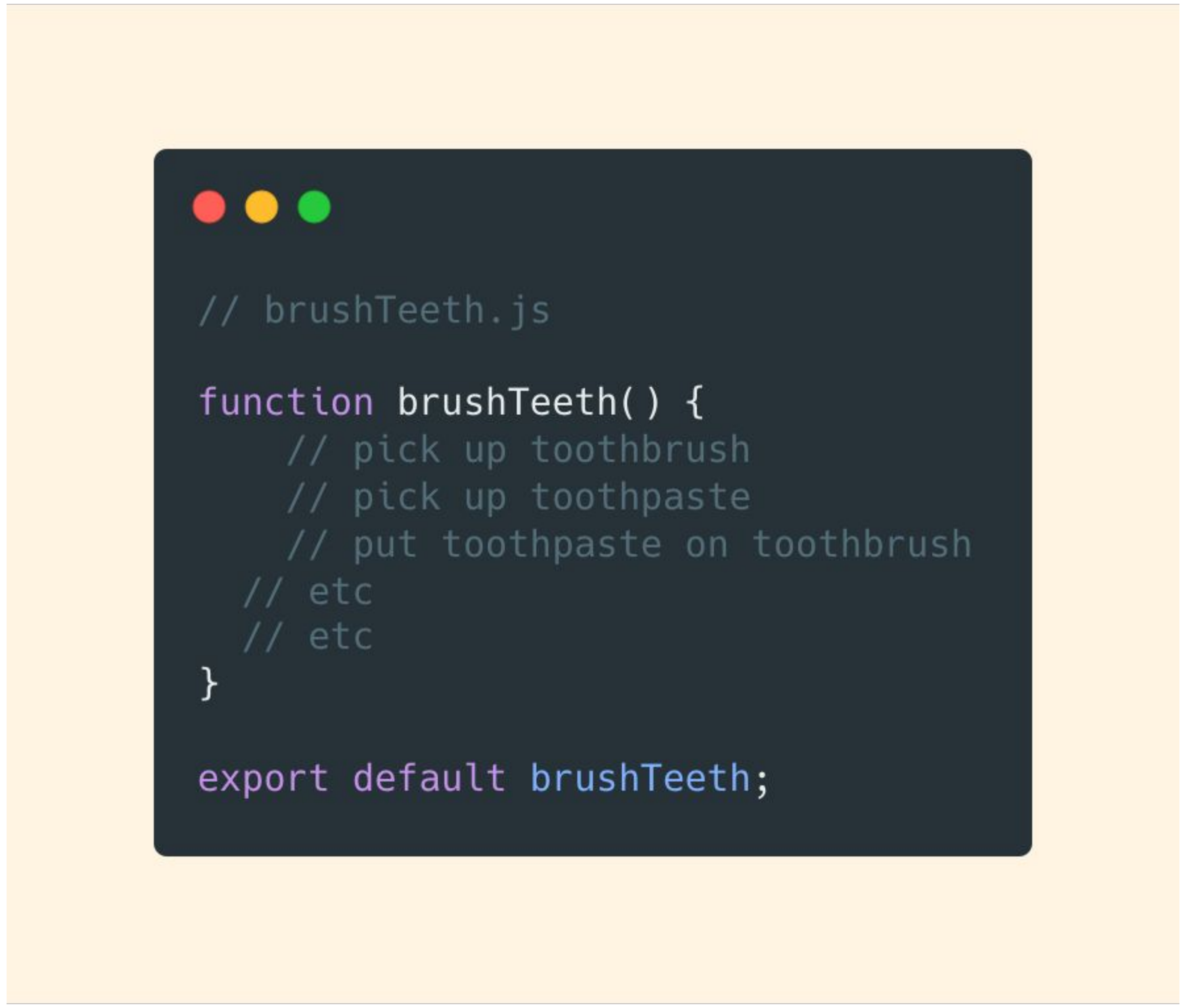


```
index.html
src/
    index.js
    modules/
        brushTeeth.js
        skincare.js
        putOnClothes.js
        sortOutHair.js
        turnOffLights.js
        getIntoBed.js
        closeEyes.js
        sleep.js
```

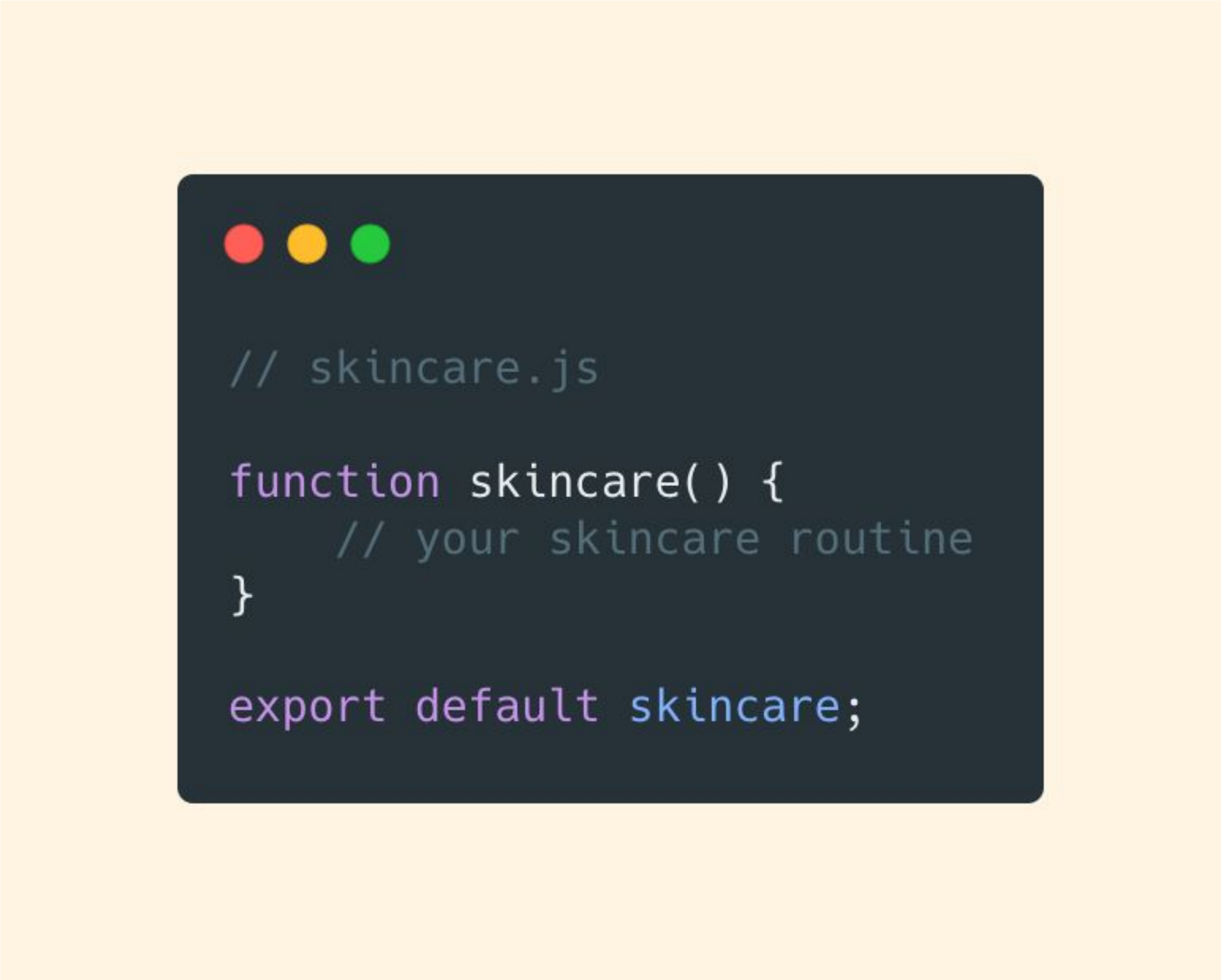
- `src` is shorthand for *source* and is where all our source files live.
- `index.js` is point of entry. Think of it as border control
- `modules/` is the folder where all our modules live

HOW TO USE MODULES

In the following examples you will see that each module is exported, and finally they are imported in the `index.js`

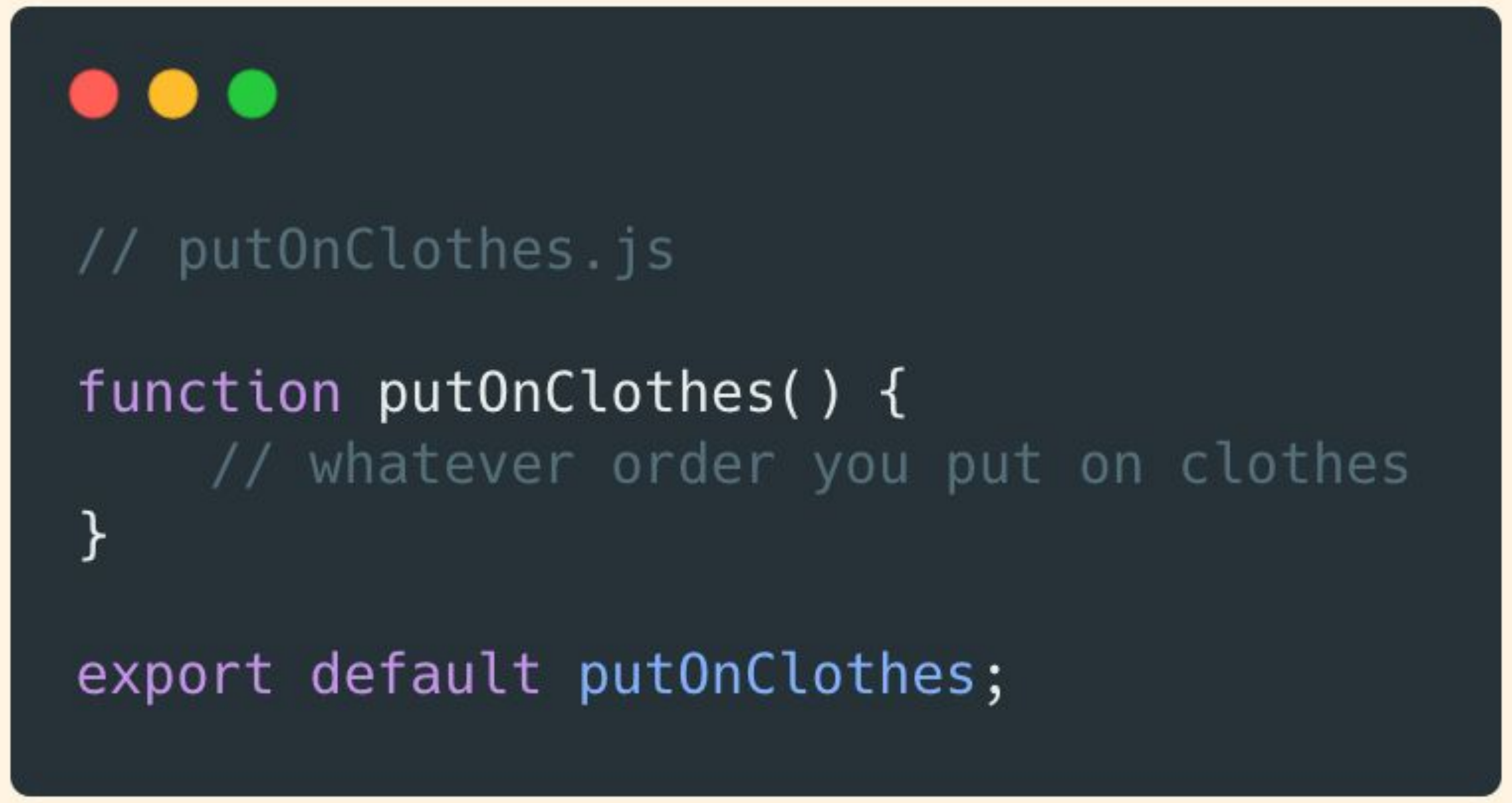


```
// brushTeeth.js

function brushTeeth() {
    // pick up toothbrush
    // pick up toothpaste
    // put toothpaste on toothbrush
  // etc
  // etc
}

export default brushTeeth;
```

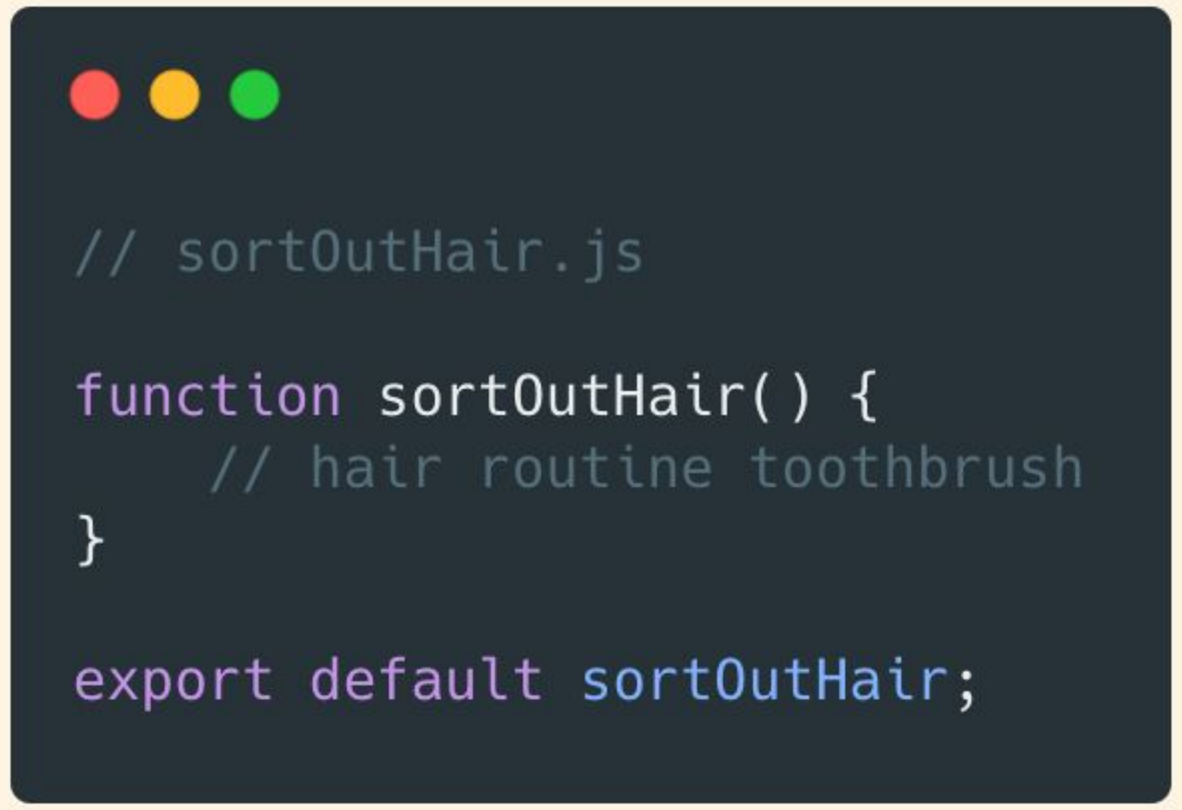


```
// skincare.js

function skincare() {
    // your skincare routine
}

export default skincare;
```

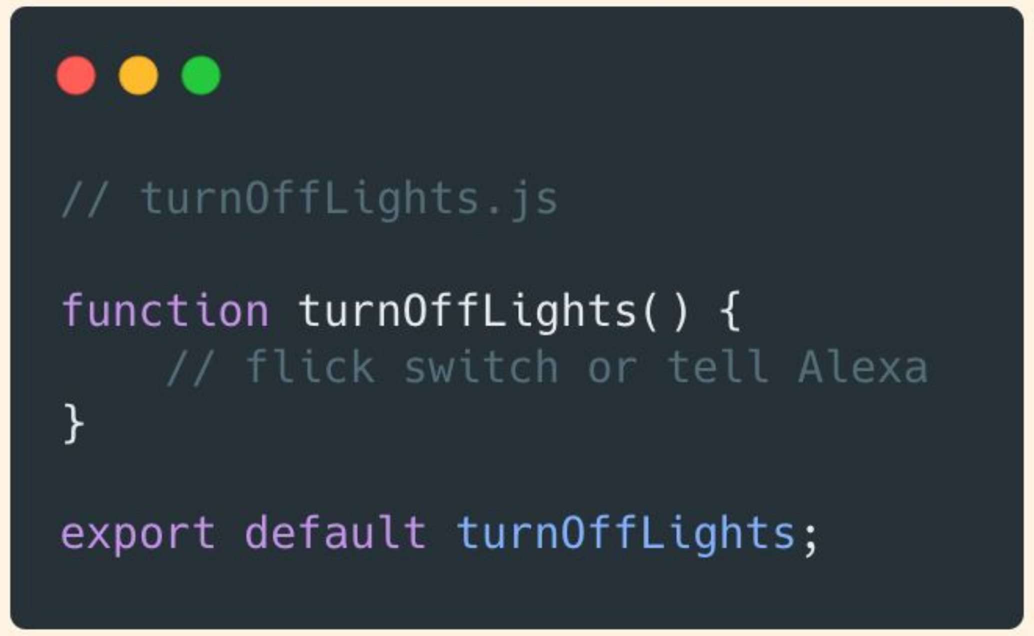
HOW TO USE MODULES



```
// putOnClothes.js

function putOnClothes() {
    // whatever order you put on clothes
}

export default putOnClothes;
```



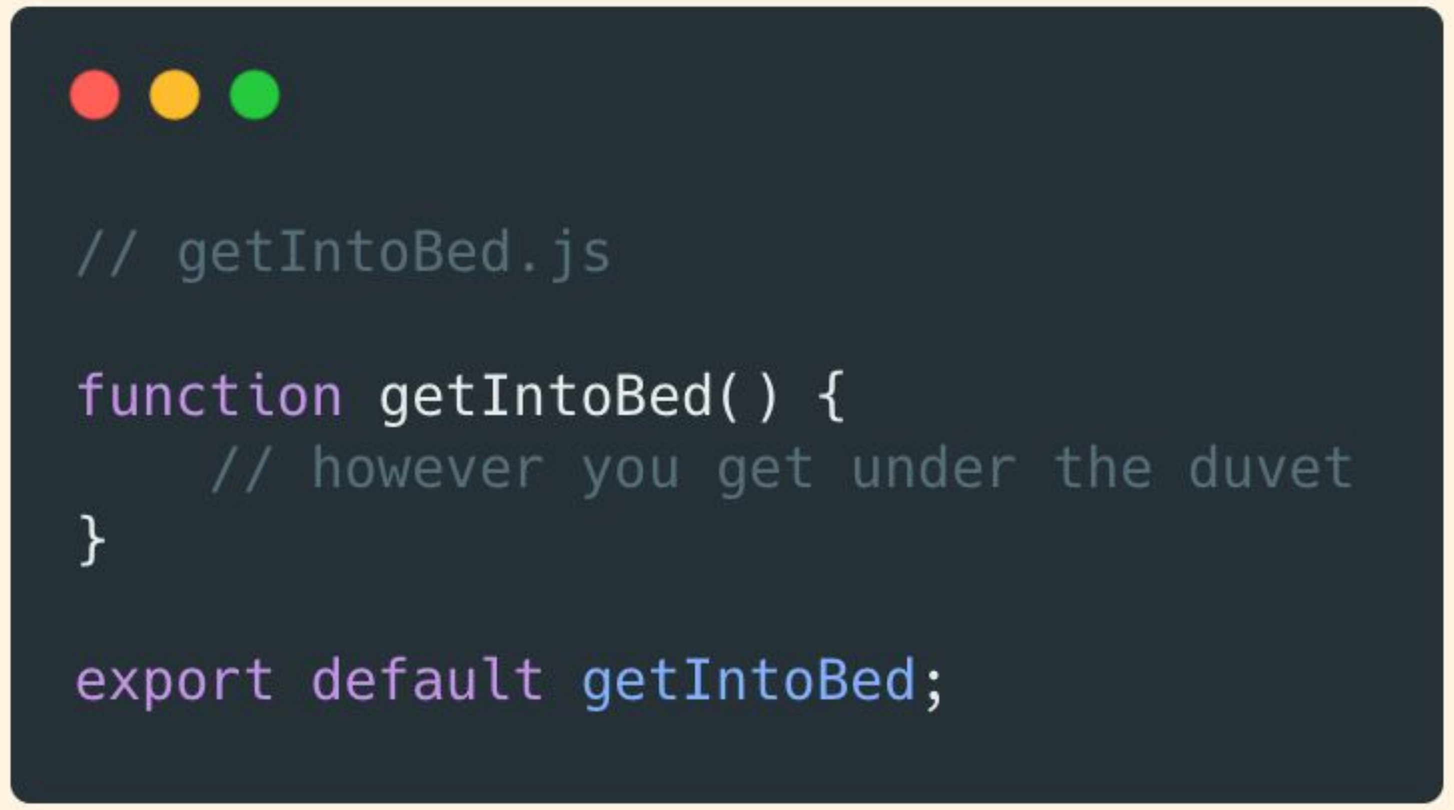
```
// sortOutHair.js

function sortOutHair() {
    // hair routine toothbrush
}

export default sortOutHair;
```



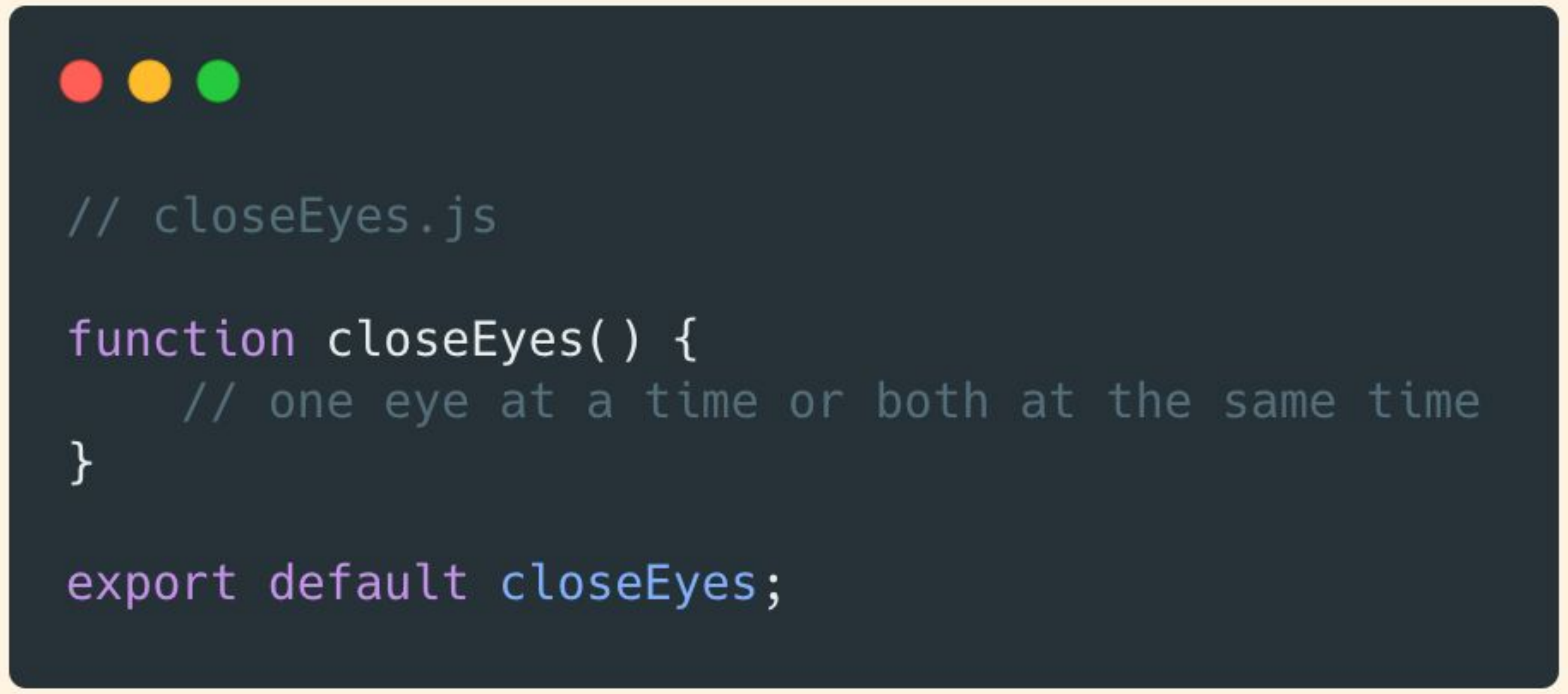
```
// turnOffLights.js

function turnOffLights() {
    // flick switch or tell Alexa
}

export default turnOffLights;
```

HOW TO USE MODULES

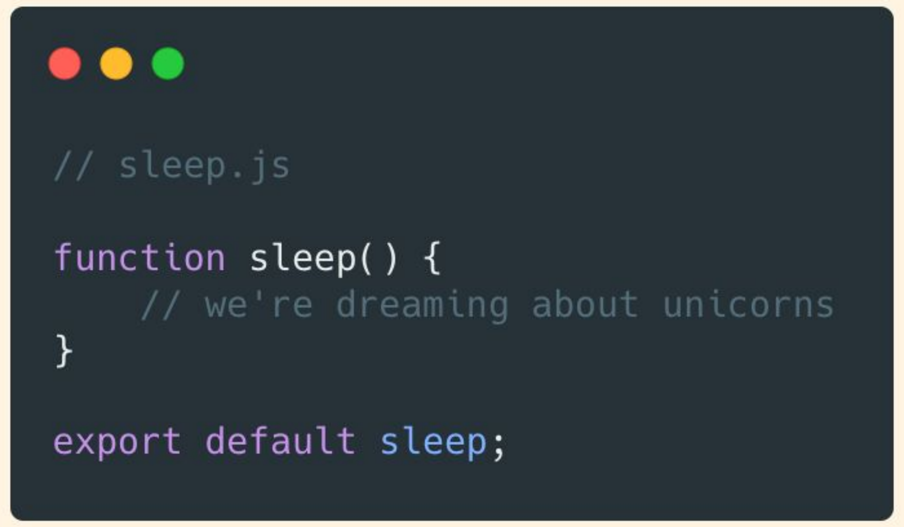
BLACK CODHER



```
// getIntoBed.js

function getIntoBed() {
    // however you get under the duvet
}

export default getIntoBed;
```



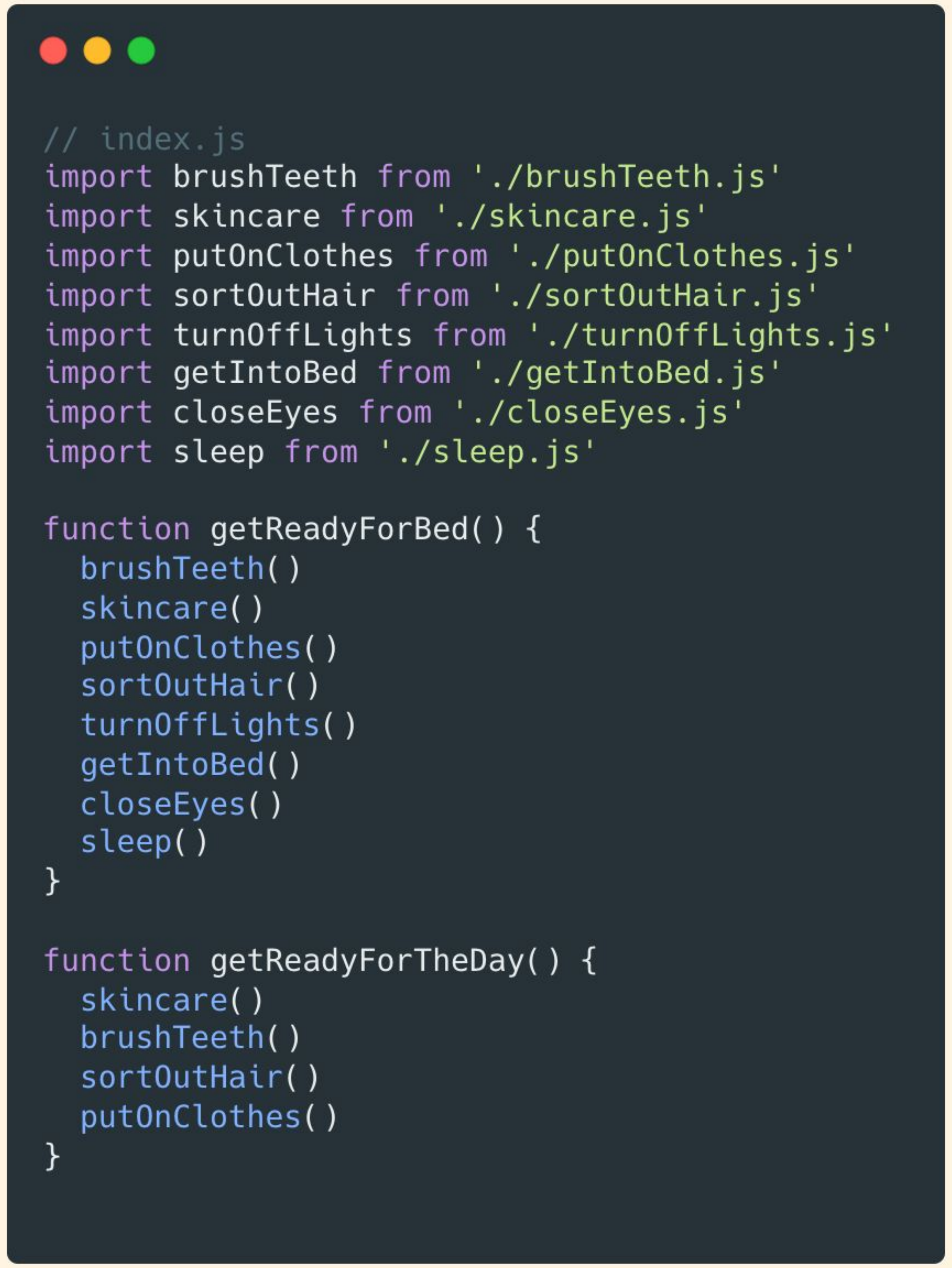
```
// closeEyes.js

function closeEyes() {
    // one eye at a time or both at the same time
}

export default closeEyes;
```



```
// sleep.js

function sleep() {
    // we're dreaming about unicorns
}

export default sleep;
```

HOW TO USE MODULES

BLACK CODHER



```
// index.js
import brushTeeth from './brushTeeth.js'
import skincare from './skincare.js'
import putOnClothes from './putOnClothes.js'
import sortOutHair from './sortOutHair.js'
import turnOffLights from './turnOffLights.js'
import getIntoBed from './getIntoBed.js'
import closeEyes from './closeEyes.js'
import sleep from './sleep.js'

function getReadyForBed() {
  brushTeeth()
  skincare()
  putOnClothes()
  sortOutHair()
  turnOffLights()
  getIntoBed()
  closeEyes()
  sleep()
}

function getReadyForTheDay() {
  skincare()
  brushTeeth()
  sortOutHair()
  putOnClothes()
}
```

You'll notice that in `getReadyForTheDay` the order of the reused modules differs from `getReadyForBed`

Checkpoint!

BLACK CODHER

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

JAVASCRIPT PROJECT

JAVASCRIPT PROJECT

You may remember from the very first JavaScript session that one of the learning outcomes for this unit is to build a multiple choice quiz to add to your portfolio. Well the time has come to start building it.

The purpose of this quiz is to reinforce what has been taught over the past few weeks.

You can use **variables**, **functions**, **arrow functions**, **arrays**, **objects**, **loops, modules** and more.

JAVASCRIPT PROJECT

The quiz should consist of 3-5 questions, with different choices for the player to choose from.

Possible quiz subjects:

- Music
- Movies
- Pop Culture
- Some of the events of 2020
- Food
- General Knowledge
- Any Buzzfeed quiz subject

JAVASCRIPT PROJECT

Steps you may want to consider:

- Create the HTML structure
    - A `<div>` to hold the quiz
    - A `<button>` to submit the quiz
    - A `<div>` to display the results
- Get each element and store each in a `variable`
- Create an **event listener** for when your user has clicked on the button
- Create an **array of objects** with your questions, options for the answer and the actual answer

JAVASCRIPT PROJECT

- Create a `buildQuiz` function
  - loop through the array of questions to display the questions and options for the answer
- Create a `showResults` function
  - keep track of how many answers the user has correct
  - check if what they selected is correct
  - change the styling if the answer right or wrong
- Put the whole thing!

JAVASCRIPT PROJECT

- Create a `buildQuiz` function
  - loop through the array of questions to display the questions and options for the answer
- Create a `showResults` function
  - keep track of how many answers the user has correct
  - check if what they selected is correct
  - change the styling if the answer right or wrong
- Put the whole thing!

AND FINALLY

- Ask for help if you need it! We’re here for you 🤗
- Push your progress to GitHub (make small commits of every bit tiny bit of progress you make)
- Good luck
- See you in React ❤️